Alternative Approaches to Mission Control Automation at NASA's Goddard Space Flight Center

Miranda Cooter, George Davis, Jennifer Mackey
Commerce One Global Services
1100 West Street
Laurel, MD 20707
Contact: **Miranda.cooter@commerceone.com**

Michael Rackley
Code 581, Systems Integration and Engineering Branch
NASA Goddard Space Flight Center
Mailstop 581.0
Greenbelt, MD 20771

Abstract

To meet its objective of reducing operations costs without incurring a corresponding increase in risk, NASA is seeking new methods to automate mission operations. This paper examines the state of the art in automating ground operations for space missions. A summary of available technologies and methods for automating mission operations is provided. Responses from interviews with several space mission Flight Operations Teams (FOT) to assess the degree and success of those technologies and methods implemented are presented.

Mission operators that were interviewed approached automation using different tools and methods resulting in varying degrees of success – from nearly completely automated to nearly completely manual. Two key criteria for successful automation are the active participation of the FOT in the planning, designing, testing, and implementation of the system and the relative degree of complexity of the mission.

Introduction

To reduce cost and manpower, future NASA space missions, especially those involving multiple spacecraft such as constellations and formations, are seeking to automate mission operations. Some progress has been achieved in automating Mission Operations Control Centers (MOCC) over the past several years. Systems controlled and monitored in a MOCC include the spacecraft bus, payload instruments, command and control systems, ground stations, communication networks and data systems. These systems must function 24 hours per day and must be monitored either by a person or by a computer. In principle, the more automated the functions, the more cost-effective operation of the remote system becomes.

This paper examines the current state of the art in automating ground operations for space missions. First, a summary of available technologies and methods for automating mission operations at GSFC is provided. Next, responses from interviews with the FOTs from several missions at NASA are presented. These interviews were conducted to assess the degree of automation implemented in their respective space missions, to study each unique approach, and to assess their relative degrees of success. Finally, a recommendation for future work in automating mission operations is provided.

Automatic Systems for Monitor and Control

Methods currently available for automating a MOCC are summarized below. Some are in use at the present, while others are in the development phase. All are loosely based on three approaches: Rule-Based Expert Systems, Procedural/Scripting Based Systems and Finite State Modeling.

Rule-Based Expert Systems. Rule-Based Expert Systems capture the knowledge of an expert and define rules to apply this information in a given situation or problem. In this case, they are used to duplicate operator interaction with the spacecraft. At the heart of the expert system is an inference engine, which matches the data to the appropriate response or rule.

Several expert system shells have real-time applications to mission operations. These include RT Works, G2/IMT and GenIE/GenSAA. All three have been considered for application to GSFC missions [Bane 1996]. All three consist of graphical interfaces with which to build expert systems as well as an interface for real-time data. GenIE/GenSAA was developed at GSFC and has been successfully applied on a number of missions. Neither RT Works nor G2/IMT has had much use at GSFC likely due to the success of GenIE/GenSAA.

Generic Inferential Executor (GenIE) is a graphical tool that allows a C Language Integrated Production Shell (CLIPS) expert system to automate real-time spacecraft pass operations, including commanding. GenIE applications duplicate the routine monitoring, decision-making, and actions of FOT personnel. GenIE is an extension of Generic Spacecraft Analyst Assistant (GenSAA), which was developed as a mission operations tool for creating the knowledge base and was demonstrated at GSFC in the early 1990's. For more information see URL: **http://aaaprod.gsfc.nasa.gov/gensaa/**.

GenSAA/GenIE was initially chosen for use in the Automated Payload Operations Control Center (APOCC), which was designed for use with the Transportable Payload Operations Control Center (TPOCC) command and control system developed at GSFC in the mid 1990's. APOCC was developed for the Extreme Ultra-Violet Explorer (EUVE) mission to prepare for its move to the University of California, Berkley [LMSMSS 1997]. The use of GenIE/GenSAA in several missions, as will be discussed in the interview section, was based on the APOCC approach.

GenSAA allows users to create highly graphical mission health and safety monitoring and fault isolation systems without requiring any source-code programming in CLIPS. A graphical user interface (GUI) is used to design and build a model of the system (e.g., spacecraft subsystems), allowing the user to define from simple to complex rules for how GenSAA should interpret and react to incoming data. GenSAA also provides a data acquisition component, which allows the expert system to interact directly with the TPOCC data server.

To support GenIE, new features were implemented in GenSAA that allow applications to act by invoking System Test and Operations Language (STOL) directives and procedures. Thus GenIE is capable of interactively sending commands to a spacecraft through TPOCC. For automated mission operations, a script that duplicates the routine monitoring, decision-making and actions of flight operations personnel controls a GenIE application. Pass scripts are individual tasks that are organized in a flowchart-like structure. This structure represents the sequence of time-constrained decisions and actions, plus background monitoring activities, which are performed by flight operations teams during a pass.

Procedural or Scripting Based Systems. A procedural or script-based system differs from an expert system in that there is no knowledge-base structure or intelligence inherent in the system. Instead, rules are established which are directly matched to a single action. Only one rule can be applied at a time and only one point can be examined at any given instant by a script. Typically these systems are highly mission-specific and are developed specifically for that particular command and control system.

Many spacecraft have based their approach to automation on procedures or scripts written in languages like STOL or Spacecraft Control Language (SCL). These scripts and procedures contain rudimentary “if-then” rules for operating the spacecraft.

STOL is the most common scripting language used in MOCCs at GSFC. Advanced Spacecraft Integration and System Test (ASIST), TPOCC, and Integrated Test and Operations System (ITOS) all provide versions of STOL that can call UNIX shells and PERL scripts directly. One version of ASIST, developed for the Imager for Magnetopause-to-Aurora Global Exploration (IMAGE) mission, permits multiple STOL procedures to run unassisted. Of all the commercial of the shelf (COTS) tools available, only SCL has an on-board interpreter.

SCL includes an on-board element that can interpret higher lever commands and, through on-board rules, generate detailed payload and spacecraft commands. SCL is unique among scripting languages in that it combines features of a traditional scripting language, such as STOL, and which has the logic capture capability of a rule-based expert system and the functionality of a database definition language. The SCL architecture consists of several modules that together can form the control system core. SCL, when combined with other products for displaying telemetry and a mission planning system, can be used as the primary command and control software in a control center. Functions are added by integrating with existing MOCC hardware and software components, graphical elements, other 3rd party products and new custom code. For more information on SCL see URL: **http://www.interfacecontrol.com/product.htm**.

Finite State Modeling. Finite state modeling represents a new approach to command and control systems for mission operations. Its approach to automation relies upon the creation of models representing the various states of a spacecraft component that are computed based on telemetry data. Component states can then be used to define subsystem states, and ultimately the spacecraft state. In this manner, various high-level spacecraft states (e.g. maneuvering, communicating or observing) can be represented by vectors of low-level component states and their scalar magnitudes. Transitions between states can be defined by linear mappings whose matrix inverses implicitly define the commands needed to enact the transitions. In this manner, finite state modeling can be used to automate spacecraft commanding. Two tools permitting state modeling currently exist at GSFC – Altair and a new version of GenIE.

At a high level, Altair has a state description facility that lets engineers and investigators describe and name spacecraft states in terms of telemetry variable ranges and tolerances; a state detection engine uses these descriptions to classify what the craft is currently doing. At a lower level, Altair has its own commanding language which is a set of extensions to the UNIX csh shell. Thus, like STOL, Altair’s commanding language can interface directly with the underlying UNIX operating system to activate scripts. For more information about Altair, see URL: **http://www.altair.com**.

Altair is being considered or has been considered for several missions at GSFC. An Altair system is being developed as a prototype for the MAP (Microwave Anisotropy Probe) mission, but at this writing there are no plans are to implement Altair as MAP’s primary command and control system. Also, a test of Altair’s concept is being devised for an experiment on the WIRE spacecraft.

Mission Interviews

Interviews were conducted with FOT members of nine current missions to assess the degree and success of automating operations. Responses are summarized in Table 1. Note that MAP is not yet operational. It can be seen that automation yielded dramatic reductions in mission operations staff on the GRO, RXTE and IMAGE missions of up to a factor of 3. None currently has “roundtrip" automation. One mission, RXTE, is capable of routinely transmitting stored command loads automatically and only IMAGE automates its mission planning. The follow sections summarize each mission's approach to automation and assess their respective success.

MAP/IMAGE. IMAGE and MAP are two spacecraft that will eventually share the same control center and the same operational team. Both systems embraced automation early in their development and included the FOT in the concept development and as a part of the Integration and Test (I & T) team. Critical components of the automated control center were written and tested by FOT members during spacecraft I & T.

IMAGE was launched in March of 2000. By August of that year, its control center was operating "lights out", i.e., operators and engineers staff the control center only during the nominal daylight shift. The IMAGE spacecraft has no propulsion system and an uncomplicated navigation system, which simplifies Mission Planning and command load generation when compared to more complex missions like GRO, RXTE and Terra.

A routine commanding system for verification and retransmission of science data was developed by the FOT. Science data downlink verification is automatically checked by the system to verify that packets have been received. The system generates a table of commands to send to the spacecraft to downlink missing blocks of data. The ASIST command and control software was upgraded for IMAGE to permit execution of simultaneous STOL procedures to monitor data. The Satellite Emergency Response System (SERS) is used to page operators when there are problems. SERS provides information on the nature of the problem by using event messages from ASIST. Remaining manual tasks for the IMAGE FOT include verifying and uplinking stored command sequences, scheduling Deep Space Network (DSN) supports and reviewing trend data.

The MAP mission will build upon IMAGE's success and further the path towards full automation. MAP will use the

Table 1 – Summary of Project Automation for GSFC Mission

NA= Not Applicable; -- = No data available; NYL = Not Yet Launched

	Project	MAP	IMAGE	GRO	RXTE	SMEX	LANDSAT 7	FUSE	TERRA	HST
	Launch Date	2Q 2001(est.)	June 2000	April 1991	December 1995	Various	April 1999	June 1999	December 1999	April 1990
	Date Automated	NYL	August 2000	1996	1997	Spacecraft Dependent	4Q 2000	NA	NA	NA
	Time to Automate	NYL	System Designed to be Automated	2 years	2 years	Spacecraft Dependent	1 year	NA	NA	NA
	Staff prior to automation	12 (Launch Operations)	8 (Launch Operations)	22	15	--	--	13	10+	30+
	Staff after automation	3 (est.)	2.5	7	7	--	--	NA	NA	NA
	Automation Tool	STOL Procedures	STOL Procedures	Expert System	Expert System	STOL Procedures	STOL Procedures	SCL for Instrument	None	None
	Ground Station	DSN	DSN	TDRSS	TDRSS	Poker Flats, Wallops & McMurdo	Various sites managed by Wallops	Puerto Rico, Hawaii, DSN & others as needed	TDRSS	TDRSS
Functions Automated	Spacecraft Monitor	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Functions Automated	Routine Spacecraft Commanding	Yes	Yes	Yes	Yes	No	No	No	No	No
Functions Automated	Uplink of Stored Command	Yes	No	No	Yes	No	No	No	No	No
Functions Automated	Mission Planning	Yes	Yes	No	No	No	No	No	No	No
Functions Automated	Command Sequence Generation	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes
Functions Automated	Ground Station Control	NA	NA	NA	NA	NA	No	Yes for Puerto Rico	NA	NA

Mission Operations Planning System (MOPS), an Oracle-based planner, in concert with ASIST to verify and transmit both routine commands for data downlink as well as stored command sequences. In contrast to IMAGE, MAP has a propulsion system. The MOPS system in concert with an FOT developed automated navigation system will execute and monitor spacecraft maneuvers. MAP's system automates all routine real-time and mission planning functions; plotting and reviewing trend data remains the sole manual task.

The success of IMAGE and MAP in automating mission operations lies in the integration of the FOT into the I & T teams, and in the simplicity and robustness of the spacecraft design. FOT members were certified as Test Conductors at the factory and tested many of the automation methods used on-orbit along with other spacecraft systems. This process permitted IMAGE to go "lights" out" in a record 3 months after launch. IMAGE would actually have been automated sooner were it not for a failure on-board, requiring a major work-around and software patch.

MAP and IMAGE are relatively simple missions from a mission-planning standpoint and, since they are survey missions, a data loss can be tolerated. For missions where data collection translates directly into dollars or where unique and timely data must be collected at any cost, this simple approach might be less applicable.

ALTAIR was considered for the MAP system and, in fact, continues to develop components for the system. It is expected that someday the Altair state engine will, as a minimum, execute in the background, but further integration with MAP systems has been delayed until after launch.

GRO/RXTE. The Gamma Ray Observatory (GRO) was on-orbit from April 1991 through June 2000. Upon completion of its initial 2.5-year and subsequent 5-year extended mission, funding reductions rendered automation a necessity.

The GRO system was developed using a very small team of developers co-located in the GRO Payload Operations Control Center (POCC) with FOT personnel where they worked side-by-side to develop the rules for the expert system. GenSAA/GenIE was used to set up the CLIPS expert system to monitor state-of-health data and GenIE was used to send routine commands and to notify operators in case of anomalies. Various UNIX and PERL scripts were used to monitor ground systems when operators were not present. With the exception of the transmission of the stored command load and Mission Planning, the GRO system was fully automated. GRO was a fairly complex mission with pointing requirements and four on-board instruments and it used the Tracking and Data Relay Satellite System (TDRSS) network. Mission planning functions were thus more complicated than for MAP or IMAGE.

The Rossi X-ray Timing Explorer (RXTE) was launched in December 1995. During spacecraft deployment, RXTE's Solar Arrays developed a crack. While the solar arrays function for routine data collection, extended periods of direct exposure to sunlight could end the mission. Thus, a quick response time is essential to the RXTE mission. Accordingly, RXTE is staffed 16 hours per day seven days a week even though this mission is highly automated and would otherwise be capable of lights-out operation. However, the controllers have assumed the responsibility for mission planning as a result of the automation. This is an example of how automation can increase staff efficiency.

RXTE uses the TPOCC command and control software and so adopted the GenSAA/GenIE approach. RXTE implemented an automated control center similar to GRO. As GRO was able to leverage lessons development performed for APOCC, so RXTE was able to leverage development performed for GRO. At this writing, a similar approach for the ACE mission is under development.

SMEX. The Small Explorer (SMEX) missions share a POCC and have common ground systems. They all use ITOS as their command and control system, two (and occasionally three) ground stations (Poker Flats, Wallops and McMurdo), and in-house mission planning systems.

STOL procedures within ITOS are used to monitor the system when an operator is not present. Like ASIST, ITOS STOL can call UNIX and PERL scripts directly permitting full ground system control. These STOL procedures are developed and implemented by the FOT. SERS is used to page the FOT when off-nominal conditions are detected. In general, however, these procedures send only routine commands including commands required for science data collection. Stored command loads are still transmitted manually.

Two SMEX missions were fully automated with the exception of Mission Planning for testing purposes and, in one case, in preparation for delivery to another site. The WIRE mission was operated for nearly three months automatically. The Fast Auroral Snapshot Explorer (FAST) system was also operated lights-out for three months prior to its delivery to the University of California, Berkley. Automation for WIRE and FAST was accomplished using the ITOS version of STOL and combined with UNIX and PERL scripts to control ground data flow.

FUSE. The Far Ultraviolet Spectroscopic Explorer (FUSE) mission, launched in June 1999, is operated by John Hopkins University's (JHU) Applied Physics Laboratory (APL) in Baltimore, Maryland. ASIST was used for the spacecraft in I & T and SCL was used for the Instrument. SCL was combined with SAMMI, a telemetry display tool used by ASIST, for use in the MOCC. This combination necessitated translating spacecraft scripts required for mission operations from STOL to SCL prior to launch. Choosing Sammi as the telemetry display tool reduced the amount of time required to create telemetry display pages.

The FUSE team built a ground station at the University of Puerto Rico as its primary ground station. This ground station is fully automated including scheduling and monitor tasks. It can be manually controlled from the MOCC at JHU if necessary. FUSE operations remain largely manual requiring the presence of two operators 24 hours a day 7 days per week.

Although it was used during I & T only for the instrument, SCL is used for all commanding in on-orbit operations. However, only the instrument makes use of SCL's on-board capabilities that permit a higher level of commanding due to its on-board interpreter and resident scripts. The spacecraft has a traditional stored command system wherein commands to perform housekeeping functions such as loading a state vector for navigation or turning on a transmitter are time-tagged and stored on-board directly.

The FOT was not involved in the I & T phase and only became involved late in the mission development phase. Thus the focus of the FOT was to prepare the MOCC for launch and routine mission operations rather than to automate mission operations functions. The FUSE mission operations team is currently considering ways of automating their control center in preparation for their extended mission, which begins in late 2002.

Landsat 7. The Landsat 7 satellite was launched in April 1999. The TPOCC system is used for command and control. Recently, the FOT developed a means of automating spacecraft monitoring to enable un-staffed operations in the MOCC during off-shifts. This scheme, called Landsat On-Orbit Flight Automation (LOOFA), uses STOL procedures to configure for contacts and to check incoming telemetry data. Scripts are also executed to monitor the status of the MOCC hardware and software. The system automatically pages an operator when anomalous telemetry or system values are detected.

A key enabling factor in this approach is that no other functions such as commanding or recorder management are required on the off-shifts. This simplifies the task required of automation since commanding, mission planning and ground station maintenance are not addressed. While other Landsat 7 mission operations are not automated, development and implementation of the LOOFA system by the FOT eliminates routine off-shift operations with minimal development costs.

Terra. Launched December 1999, Terra's original mission operations concept envisioned a highly automated ground system. One year prior to launch it became clear that the automated system would not work and a replacement was quickly found. This change limited development time for the mission operations system and resulted in a system that, while fully functional, does not provide the integrated automated system originally envisioned. The ECLIPSE system is used for satellite command and control. A separate real-time control system, Epoch 2000, is used to communicate with the Network Control Center (NCC). EPOCH 2000 and its partner tool, Archive Browser and Extractor (ABE), are used to provide offline analysis and trending support. Terra employs a mission-specific application for mission-planning support based referred to as the Mission Management System (MMS).

Unlike simpler missions such as MAP, IMAGE and the SMEX series of spacecraft, Terra has four instruments on board and requires detailed mission planning. ECLIPSE does not yet have the capability to interact with the NCC for monitoring and configuring TDRSS support, so the FOT must use the separate Epoch 2000 for that purpose. This problem will be corrected in a future delivery, however this is one example of the manual nature of Terra's mission operations.

Terra is staffed with four operators 24 hours per day 7 days per week to operate the satellite and ground systems. While a few areas have been semi-automated, such as the generation of trending plots and reports, in general Terra operations are not yet automated. Terra’s FOT is currently investigating ways of automating their mission operations to reduce manpower and cost.

HST. The Hubble Space Telescope (HST) mission operations system was also developed at GSFC. HST, launched in April 1990, is unique among missions in that once every three years, HST requires a Shuttle-based manned mission for servicing. This fact alone changes the nature of mission operations and requires that a knowledgeable and well-trained staff be available to support and plan the servicing missions. Thus there is limited motivation to implement full automation.

HST has reduced its console staff (24 hours per day, seven days per week) to 4 per shift, however it is not expected to reduce the staff much further until after the last servicing mission when they expect to implement some form of full automation. Their latest command and control system includes provisions for implementing a GenIE/GenSAA based system. HST has, however had some success in automating mission planning functions.

Conclusions

Clearly, mission operations automation at GSFC has made great progress in the last five years. Many missions can now monitor telemetry unattended, send routine commands and notify operators regarding anomalous states. However, very few missions have automated their mission planning or commanding processes and none have fully automated the trending process.

All missions in which automation has been highly successful have included the FOT in the development process. The most successful, IMAGE, included the FOT in the integration and test process which produced not only an extremely knowledgeable launch team, but also a system which has been very successful in a short time. It is expected that MAP will build on IMAGE's success and will likely be the first fully automated control system requiring no routine interaction by operators.

There are no apparent technological barriers to implementing a highly automated MOCC. Full support and participation by the FOT is paramount to automation success regardless of the tools used to implement the automation. The earlier the FOT is involved and the earlier mission operations are addressed in the system design, the more likely an automated MOCC will result. Both the MAP and IMAGE missions have taken this approach and are currently the most successful missions.

Each mission has taken a unique approach to automating mission operations. On extended missions such as GRO and RXTE on which a team of flight controllers developed a definable expertise prior to automation, the expert system approach has been highly successful. For missions in the development phase, inclusion of the FOT can insure a successfully automated MOCC as demonstrated by MAP and IMAGE.

The ease with which a system can be automated depends on the mission itself and, in particular on its mission planning tasks. Additionally, the cost of automating mission operations must be weighed against predicted mission life, risk to the spacecraft or ground station if response to an anomaly is delayed, required delivery time for science data, and cost (in both dollars and understanding) of a data loss. Only missions with very simple mission planning tasks have succeeded in automating the mission planning phase and the resulting command upload phase. Those systems requiring detailed mission planning are thus, by their nature, more difficult to automate. In order to be successful, these missions must look to new solutions for automation. No mission interviewed has fully automated offline trending analysis and no mission included response to even “routine” anomalies via automated commanding.

There were various approaches to automation assessed in this study, each of which has its place. It is clear that, for simple missions (e.g.. MAP and IMAGE) a high degree of automation can be achieved with procedural scripts. However, for automation of complex missions, simple STOL scripts are not sufficient. There is clearly a need for more intelligent systems (either rule- or state-based) if robust autonomy for larger, more complex, missions such as GRO and RXTE is to be achieved.

Recommendations. Based on the interviews and a study of available systems, three areas are clearly in need of future research and development with regards to fully automating mission operations. They are: 1) automation of mission planning to the extent that direct commanding can result; 2) further development of expert systems to the point where autonomous commanding in response to anomalies can occur; and 3) automation of offline analysis.

Expert systems and finite-state modeling systems should continue to develop and expand their focus to include the mission planning and commanding aspects of mission operations. Spacecraft system developers should consider building an expert system or a finite state model as a part of their overall spacecraft system design and development task.

Acknowledgements. The authors would like to express their gratitude to all the missions who participated in the interviews, particularly J.B. Joyce and Chris Silva of FUSE, Matt Fatig of GRO and RXTE, Kevin Hartnett of EUVE, Kevin Blahut of MAP, Kieth Walyus of HST and Dion Smallwood of IMAGE.

References

Bob Bane, “G2/IMT vs GenSAA/GenIE vs RTWorks/ALTAIR as VMOC Platforms”, June 21, 1996, **http://aaaprod.gsfc.nasa.gov/TEAS/june/Bbpaper.html**

LMSMSS (Lockheed-Martin Space Mission Systems and Services), “As-Built System Description Document for the Extreme Ultra-Violet Explorer (EUVE) Automated Project Operations Control Center (APOCC) Prototype”, July 1997

Ed Luczak, NASA GSFC Code 522, Lessons Learned from the Generic Inferential Executor (Genie) Project, DSTL-96-007, April 1996. **URL:http://aaaprod.gsfc.nasa.gov/gensaa/genie/GenieLLDocFinal.html**

Peter M. Hughes et al, “Advancing Satellite Operations with Intelligent Graphical Monitoring Systems”, AIAA Conference 1993

Gary Riley, “CLIPS: A Tool for Building Expert Systems”, August 8, 1998, url: **http://www.ghgcorp.com/clips/WhatIsCLIPS.html**

Jim VanGassbeck, Allan Posner, Brian Buckley, “Distributed Space-Segment Control Using SCL”, 1998, url: **http://www.interfacecontrol.com/white.htm**